Implementation and Algorithms for Vertex-DSM-Tree version of February 11, 2004

Falk Meissner

February 11, 2004

I start with the Vertex DSM below, to document whats available in the Last DSM. For details how those bits get there refer to the other sections below. For a list of scaler bits available from the vertex tree for the lumi/asy/FPD scaler see Section11.

1 Vertex DSM L1-VT201, layer2

All threshold bits of the Vertex tree are brough into the Vertex DSM. But only the first threshold comparision (th0) are given to the last DSM, since for BBC/ZDC only one threshold bit is reserved in the TCU. (th0 and th1 are independent and can be set to any value) The other threshold bits are here for easy later adjustments, all threshold comparision end up in some scaler output.
Note: The inclusion of the Tof causes some changes which are marked below

Inputs: From small tile BBC-DSM BB101
(0-8) TAC-Difference;
(12/13) Small-ADC-East/West> th0
(14/15) Small-ADC-East/West> th1

From Large tile BBC-DSM BB102
(0/1) Large-ADC-East/West > th0
(2/3) Large-ADC-East/West > th1
(15-12) Quad-Hit-Map, ADC-Large T/B/N/S (East+West)

From ZDC DSM ZD101
(0) zdc1>ADC-th0E * dead time
(1) zdc2>ADC-th0W * dead time
(2) zdc1>ADC-th1E * dead time
(3) zdc2>ADC-th1W * dead time
(4) time1-window-E * dead time
(5) time2-Window-W * dead time
TOF
(6) tof-bit
noTOF
(6) att. Sum > th * dead time
(15-7)tac-diff

Registers: **L1-index:0x19**
(two independent timing windows for BBC and ZDC)
R0: BBC-DeltaTMin0 (9)
R1: BBC-DeltaTMax0 (9)
R2: BBC-DeltaTMin1 (9)
R3: BBC-DeltaTMax1 (9)
R4: ZDC-DeltaTMin0 (9)
R5: ZDC-DeltaTMax0 (9)
R6: ZDC-DeltaTMin1 (9)
R7: ZDC-DeltaTMax1 (9)

LUT: 1:1

1st Clock: BBC-DeltaTMin0 < BBC-Tacdiff < BBC-DeltaTMax0 →BBC-tac0
BBC-DeltaTMin1 < BBC-Tacdiff < BBC-DeltaTMax1 →BBC-tac1
ZDC-DeltaTMin0 < ZDC-Tacdiff < ZDC-DeltaTMax0 →ZDC-tac0
ZDC-DeltaTMin1 < ZDC-Tacdiff < ZDC-DeltaTMax1 →ZDC-tac1

2nd Clock empty/delays/mapping

Output: **THIS IS WHATS AVAILABLE IN THE LAST DSM!**
(0) BBC-tac0 // tac windows
(1) ZDC-tac0
(2) SmallTile-ADC-th0-East // Small Tile BBC ADC th0
(3) SmallTile-ADC-th0-West
(4) LargeTile-ADC-th0-East // Large Tile BBC ADC th0
(5) LargeTile-ADC-th0-West
(6) ZDC-ADC-th0-East; // ZDC ADC th0
(7) ZDC-ADC-th0-West;
(8) ZDC-TAC-win-East; // ZDC Tac window
(9) ZDC-TAC-win-West;
NoTOF
(10) ZDC-attSum-th0; // ZDC att Sum
TOF
(10) tof-bit //tof-trigger
(11) spare
(15-12) LargeTile-quadHitMap // Large Tile Quad-hit-map for topology triggers

Scalers: (0) BBC-tac0 // tac windows
(1) BBC-tac1
(2) ZDC-tac0*zdc-adc-th0-East*zdc-adc-th0-West*zdc-tac-win-East*zdc-tack-win-West
(3) ZDC-tac1*zdc-adc-th1-East*zdc-adc-th1-West*zdc-tac-win-East*zdc-tack-win-West
(scaler-2 = output-bits: 1*6*7*8*9, i.e. R4<zdc-tac-diff<R5, zdc-ADC-E/W>th0, separate tac windows East/West ok
(scaler-3 is the same except looking at the second thresholds R6< zdc-tackdiff<R7*zdc-

ADC>th1, separate tac windows East/West ok

2 BBC-Small-Tile East/West ADC DSM BB001-004; layer0, OLD-NO DEAD TIME

Input: 8*8bits inner PMTs ADC
8*8bits inner PMTs TAC
(Sorted by quadrants, T/B(3PMTs), S/N(5PMTs)

LUT: timing adjustments/pedestal subtraction for each PMT

Registers: **BBC; index 0x10, 0x12, 0x14, 0x16**
R0: Small-PMT-ADC-Thresh(8) R1: TdcWindowMin (8)
R2: TdcWindowMax (8)

1st Clock: for each PMT:
Timing Window: TdcWindowMin<TDC<TdcWindowMax →Good-TAC
ADC above threshold: ADC> Small-PMT-ADC-Thresh →Good-ADC

2nd Clock: Compare Good-TAC*Good-ADC →good PMT hit, for 'bad' channels ADC is set to zero (i.e. not included in the ADC sum)
Quad-Hit-Map (4 bits): Two Quads per DSM board*(inner/outer circle of small tile Annulus).
Watch: North and South PMT are swapped for East and West
PMT 1—7—8 →Top
PMT 4—12—13 →Bottom
PMT 5—6—14—15—16 →East/South or West/North
PMT 2—3—9—10—11 →West/South or East/North

Only two TAC values at a time can be compared per clock cycle. For 8 TACs, 3 cycles are needed →have to use override cycles.
If there are no good PMT hits at all, TAC quality bit is set to zero.

1st override: Sum ADC values intermediate step (sum ADC0-3 and ADC4-7) Compare 4*2TACs

2nd override: Sum ADC (0-7) Compare 2*2TACs

3rd override: Compare 2TACs →fastest TAC

4th override: Delay all outputs to get the timing right

Output: ADC-Sum (0-10), Quadrant-hit-map(11-14)(2 Quads*Inner/Outer), empty(15)
Max TAC(16-23), TAC Quality bit (24)

3 BBC-Small-Tile East/West ADC DSM BB001-004; layer0; VERSION WITH DEAD TIME

Input: 8*8bits inner PMTs ADC
8*8bits inner PMTs TAC
(Sorted by quadrants, T/B(3PMTs), S/N(5PMTs)

LUT: timing adjustments/pedestal subtraction for each PMT

Registers: **BBC; index 0x10, 0x12, 0x14, 0x16**
R0: Small-PMT-ADC-Thresh(8) R1: TdcWindowMin (8)
R2: TdcWindowMax (8)
R3: Deadtime (4)

1st Latch Inputs

2nd for each PMT:
Timing Window: TdcWindowMin<TDC<TdcWindowMax →Good-TAC
ADC above threshold: ADC> Small-PMT-ADC-Thresh →Good-ADC
delay ADC values by 2 clocks until 4th
delay TACs by 3 clocks until 1st override

3rd delay Good-TAC bits by one clock until 4th
apply deadtime to Good-ADC bits →Good-ADC+KillerBit
Instantiate deadtime components

4th define 'good hit'= adc above th and dead time ok, tac window ok
gate adc values: 'good-hit' gated ADC =input adc
all other adc's set to '0' to exclude them from adc sum
Build quadrant hit map bits from good hits
Quad-Hit-Map (4 bits): Two Quads per DSM board*(inner/outer circle of small tile Annulus).
Watch: North and South PMT are swapped for East and West
PMT 1—7—8 →Top
PMT 4—12—13 →Bottom
PMT 5—6—14—15—16 →East/South or West/North
PMT 2—3—9—10—11 →West/South or East/North

1st override: Delay quad-hit-map by 3 cycles until 4th override (output mapping)
Sum gated ADC values intermediate step (sum ADC0-3 and ADC4-7)
Compare 4*2TACs
Only two TAC values at a time can be compared per clock cycle. For 8 TACs, 3 cycles are needed →have to use override cycles.
If there are no good PMT hits at all, TAC quality bit is set to zero.

2nd override: Total ADC Sum (0-7)
Compare 2*2TACs

3rd override: Delay ADC sum by one cycle
Compare 2TACs →fastest TAC

4th override: Map output

Output: ADC-Sum (0-10), Quadrant-hit-map(11-14)(2 Quads*Inner/Outer), empty(15)
Max TAC(16-23), TAC Quality bit (24)

4 BBC-small tile BB101, layer1

Inputs: from 4 small tile ADC/TAC boards (2 East/2West)
ADC-Sum(0-10), quad-hit-map(11-14)
Max TAC+Quality (16-24).

Registers: **BBC; index:0x1a**
four registers, all thresholds can be set independently
R0: Small-ADC-Thresh0-East (11)
R1: Small-ADC-Thresh0-West (11)
R2: Small-ADC-Thresh1-East (11)
R3: Small-ADC-Thresh1-West (11)

LUT: 1:1

1st Clock: For East and West separately:
Compare 2 TACs →Fastest TAC
Sum ADC values

2nd Clock Calculated TAC-East-TAC-West →TAC-diff
Sum-ADC-East > Small-ADC-Thresh0-East
Sum-ADC-West > Small-ADC-Thresh0-West
Sum-ADC-East > Small-ADC-Thresh1-East
Sum-ADC-West > Small-ADC-Thresh1-West
Combine Quad-Hit-Maps: East/West/Inner/Outer

Output: (JP6-upper bits of output) to VT201

(0-8)	TAC-Difference
(9-11)	empty
(12-13)	Small-ADC-East/West > th0
(14-15)	Small-ADC-East/West > th1

Scalers: (JP1 lower bits of output)

(0-3)	quad-hits-East T/B/N/S
(4-7)	quad-hits-West T/B/N/S
(8/9)	hits-in-inner circle East/West
(10/11)	hits-in-outer circle East/West
(12-13)	Small-ADC-East/West > th0
(14-15)	Small-ADC-East/West > th1

5 BBC-Large-tile ADC DSM BB005,layer0 NEW VERSION with Killer bit

This module uses one override cycle!

Input: east PMT17-24, west PMT17-24
16*8bit ADC signals

Registers: **BBC; index:0x18**
R0: Large-PMT-ADC-Thresh (8 bits)
R0: Large-PMT-Deadtime (4 bits)

LUT: pedestal subtraction

1st. Clock latch 16*8 bit ADC values
delay a copy of nput until clock4

2nd Clock for each PMT:
ADC> Large-PMT-ADC-Thresh →Good Hit

3rd Clock Accept only PMT's with a good hit if there was no (good) hit for R1*RHIC strobes before (deadtime/killerbit)

4th clock Gate the actual ADC values with the good-hit-and-killerbit-bit
Combine Good PMTs per Quadrant→Quad-Hit-Map-East/West T/B/N/S

1st Override East/West separately :
Add 2*4 gated ADCs →Intermediate. ADC-Sum (adding 8 channels needs two steps). PMT's with ADC values bwlow threshold or non-zero dead-time-counter do not contribute to sum.

2nd Override East/West separately, add Intermediate. Quad-ADC-Sums → ADC-SumEast/West

3rd Override Delays

4th override Latch output

Output **2 cables; East/West:**
Large-ADC-Sum (0-10), Quad-Hit-Map(11-14), empty(15)

6 BBC-Large-tile ADC DSM BB005,layer0 OLD VERSION

Input: east PMT17-24, west PMT17-24
16*8bit ADC signals

Registers: **BBC; index:0x18**
R0: Large-PMT-ADC-Thresh (8 bits)

LUT: pedestal subtraction

1st. Clock: for each PMT:
ADC> Large-PMT-ADC-Thresh →Good Hit
East/West separately :
Add 2*4 PMT ADCs →Intermediate. ADC-Sum (adding 8 channels needs two steps)

2nd Clock: East/West separately, add Intermediate. Quad-ADC-Sums → ADC-SumEast/West
Combine Good PMTs per Quadrant→Quad-Hit-Map-East/West T/B/N/S

Output **2 cables; East/West:**
Large-ADC-Sum (0-10), Quad-Hit-Map(11-14), empty(15)

7 BBC-Large-tile DSM BB102,layer1

Input: 2 cables: East and West
Large-ADC-Sum (0-10), Quad-Hit-Map(11-14), empty(15)

Registers: **BBC; index:0x1c**
R0: LargeTile-ADC-Thresh0 East(11bits)
R1: LargeTile-ADC-Thresh0 West(11bits)
R2: LargeTile-ADC-Thresh1 East(11bits)
R3: LargeTile-ADC-Thresh1 West(11bits)

LUT: 1:1

1st. Clock: ADC-Sum East/West > LargeTile-ADC-Thresh0/1 East/West
Quad-hits-East+Quad-hits-West →Quad-Hit-Map (East/West combined)

2nd Clock: empty

Scalers: (0/1) BBC-Large-ADC-East/West th0
(2/3) BBC-Large-ADC-East/West th1
(12-15) BBC-Large-Quad Hits, T/B/N/S East+West

Output: to VT201
(0/1) BBC-Large-ADC-East/West th0
(2/3) BBC-Large-ADC-East/West th1
(12-15) BBC-Large-Quad-Hit-Map, T/B/N/S East+West

8 NoTof ZDC DSM ZD001, layer0

Input: (only the used one are listed)
zdc-adc-East(8) inp1(15-8)
zdc-adc-West(8) inp3(15-8)
zdc-tac-East(8) inp7(15-8)
zdc-tac-West(8) unp7(7-0)
zdc-attSum-East+West(8) inp4(15-8)

Registers: **BBC; index:0x1e**
(adc thresholds th0/th1 independent for East/West)
R0: zdc-adc-East-th0(8)
R1: zdc-adc-West-th0(8)
R2: zdc-adc-East-th1(8)
R3: zdc-adc-West-th1(8)
R4: zdc-tac-East-min(8)
R5: zdc-tac-West-min(8)
R6: zdc-th0-timegap(4) // deadtime for adc E/W>th0
R7: zdc-th1-timegap(4) // deadtime for adc E/W>th1

R8: zdc-timewin-timegap(4) // deadtime for tac-window
R9: zdc-attSum-th(8)
R10:zdc-attSum-timegap(4) //deadtime for attSum>th
R11:zdc-tac-East-Max(8)
R12:zdc-tac-West-Max(8)
R13: dummy, not used(4)

LUT: 1:1

1st. Clock: compare thresholds: zdc-adc*>th*
tac-timewindows: min <= zdc-Tac-East/West < max
calculate tac difference **no quality cuts here.**

2nd Clock: force deadtime for the timegap*RHIC crossings

4 overrides delay output to stay in sync with BB000-003

Output: (0) zdc1>ADC-th0E * dead time R6*104ns
(1) zdc2>ADC-th0W * dead time R6*104ns
(2) zdc1>ADC-th1E * dead time R7*104ns
(3) zdc2>ADC-th1W * dead time R7*104ns
(4) time1-window-E * dead time R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) att. SUm > th * dead time R9*104ns
(15-7)tac-diff;

9 Tof ZDC DSM ZD001, layer0

Input: (only the used one are listed)
zdc-adc-East(8)
zdc-adc-West(8)
zdc-tac-East(8)
zdc-tac-West(8)
pvpd-tac-East(8)
pvpd-tac-West(8)
tof-adc-disc.(8)

Registers: **BBC; index:0x1e**
(adc thresholds th0/th1 independent for East/West)
R0: zdc-adc-East-th0(8)
R1: zdc-adc-West-th0(8)
R2: zdc-adc-East-th1(8)
R3: zdc-adc-West-th1(8)
R4: zdc-tac-East-min(8)
R5: zdc-tac-West-min(8)
R6: zdc-th0-timegap(4) // deadtime for adc E/W>th0
R7: zdc-th1-timegap(4) // deadtime for adc E/W>th1
R8: zdc-timewin-timegap(4) // deadtime for tac-window
R9: pvpd-tacDiff-Min(9)

R10:pvpd-tacDiff-Max(9)
R11:zdc-tac-East-Max(8)
R12:zdc-tac-West-Max(8)
R13: dummy, not used(4)

LUT: 1:1

1st. Clock: compare thresholds: zdc-adc*>th*
tac-timewindows: min <= zdc-Tac-East/West < max
calculate zdc tac difference **no quality cuts here.**
calculate pvpd Tac difference

2nd Clock: force deadtime for the timegap*RHIC crossings
tof-bit= tofAdc>0 AND R9(min)< $(256 + tofTacE - tofTacW)$ <R10 There is no dead time for the tof bit.

Output: (0) zdc1>ADC-th0E * dead time R6*104ns
(1) zdc2>ADC-th0W * dead time R6*104ns
(2) zdc1>ADC-th1E * dead time R7*104ns
(3) zdc2>ADC-th1W * dead time R7*104ns
(4) time1-window-E * dead time R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) tof-bit
(15-7)tac-diff;

10 ZDC DSM ZD101, layer1

Input: 16 bit from ZD001
(0) zdc1>ADC-th0E * dead time R6*104ns
(1) zdc2>ADC-th0W * dead time R6*104ns
(2) zdc1>ADC-th1E * dead time R7*104ns
(3) zdc2>ADC-th1W * dead time R7*104ns
(4) time1-window-E * dead time R8*104ns
(5) time2-Window-W * dead time R8*104ns
NoTOF
(6) att. Sum > th * dead time R9*104ns
TOF
(6) tof-bit
(15-7)tac-diff

Registers: None

LUT: 1:1

1st. Clock: Map input to output and scaler

2nd Clock: Delay

Output: (JP6 bits 31-16 of output) to VT201 same as bit 0-15 of input

Scalers: (JP1 bits 15-0 of output) same as bit 0-6 of input (threshold bits only)

11 Scalers

11.1 Luminosity scaler

Bit	Name	From DSM	JP6 Bit
1	BBC TAC-Window 0	VT201	0
2	BBC TAC-Window 1	VT201	1
3	BBC small-ADC East > th0	BB101	12
4	BBC small-ADC West > th0	BB101	13
5	BBC small-ADC East > th1	BB101	14
6	BBC small-ADC West > th1	BB101	15
7	BBC large-ADC East > th0	BB102	0
8	BBC large-ADC West > th0	BB102	1
9	ZDC tac-diff-window0	VT201	2
10	ZDC East ADC>th0	ZD101	0
11	ZDC East ADC>th1	ZD101	2
12	ZDC West ADC>th0	ZD101	1
13	ZDC West ADC>th1	ZD101	3
14	EMC ADC¿N		
15	EMC ADC¿M		
16	CTB multi¿N		
17	CTB multi¿M		
18-24	bunch id		

11.2 BBC asymmetry scaler

Bit	Name	From DSM	JP6 Bit
1	BBC East T	BB101	0
2	BBC East B	BB101	1
3	BBC East N	BB101	2
4	BBC East S	BB101	3
5	BBC West T	BB101	4
6	BBC West B	BB101	5
7	BBC West N	BB101	6
8	BBC West S	BB101	7
9	BBC East inner circle	BB101	8
10	BBC West inner circle	BB101	9
11	BBC East outer circle	BB101	10
12	BBC West outer circle	BB101	11
13	BBC Large-ADC East > th0	BB102	0
14	BBC Large-ADC West > th0	BB102	1
15	ZDC tac-diff-window0	VT201	2
16	EMC ADC¿N		
17	CTB multi¿N		
18-24	bunch id		

11.3 FPD asymmetry scaler, TO BE REDEFINED

Bit	Name	From DSM	JP6 Bit
1	BBC TAC-Window 0	VT201	0
2	max FPD adc sum>N1		
3	max FPD adc sum>N2		
4	max FPD adc sum>N3		
5	max FPD adc sum>N4		
6	FPD East T>N		
7	FPD East B>N		
8	FPD East N>N		
9	FPD East S>N		
10	FPD West T>N		
11	FPD West B>N		
12	FPD West N>N		
13	FPD West S>N		
13	max FPD high tower id bit 1		
14	max FPD high tower id bit 2		
15	max FPD high tower id bit 3		
16	max FPD high tower id bit 4		
17	CTB multi¿N		
18-24	bunch id		

12 Tile-PMT-DSM Input Mapping

BBC scintillator tiles numbers are specified at...
http://www.star.bnl.gov/STAR/html/bbc_l/geom/front_view.html, which shows the BBC scintillator array from a vantage point that is outside of the STAR magnet toward the IP. The numbering scheme applies to the east and west sides of STAR. (Note, this viewpoint is contrary to the usual star definitions looking from the IP toward East and West.) The Tile to PMT mapping comes from Les email. The PMT to DSM channel assignments are from the trigger page cable map
http://www.star.bnl.gov/STAR/html/trg_l/TSL/Schematics/BBC_Crate_Cable_Map.html.

12.1 Small Tiles

Tile	PMT	East	Pos.	West	Ring	DSM E/W	ADC In / TAC In
1	1		T		inner	BB001/002	JP2-ch0l/JP4-ch4l
2	2	N		S	inner	BB001/002	JP7-ch1h/JP9-ch5h
3	3	N		S	inner	BB001/002	JP3-ch2l/JP5-ch6l
4	4		B		inner	BB003/004	JP2-ch0l/JP4-ch4l
5	5	S		N	inner	BB003/004	JP7-ch1h/JP9-ch5h
6	6	S		N	inner	BB003/004	JP3-ch2l/JP5-ch6l
7 9	7		T T		outer outer	BB001/002 BB003/004	JP2-ch0h/JP4-ch4h
8	8		T		outer	BB001/002	JP7-ch1l/JP9-ch5l
10	9	N		S	outer	BB001/002	JP3-ch2h/JP5-ch6h
11	10	N		S	outer	BB001/002	JP8-ch3l/JP10-ch7l
12	11	N		S	outer	BB001/002	JP8-ch3h/JP10-ch7h
13 15	12		B B		outer outer	BB003/004 BB003/004	JP2-ch0h/JP4-ch4h
14	13		B		outer	BB003/004	JP7-ch1l/JP9-ch5l
16	14	S		N	outer	BB003/004	JP5-ch6h/JP5-ch6h
17	15	S		N	outer	BB003/004	JP8-ch3l/JP10-ch7l
18	16	S		N	outer	BB003/004	JP8-ch3h/JP10-ch7h

12.2 Large Tiles

Tile	PMT	East	Pos.	West	Ring	DSM	ADC East/West In
19	17		T		inner	BB005	JP2-ch0l/JP4-ch4l
20 21	18	N N		S S	inner inner	BB005	JP2-ch0h/JP4-ch4h
22	19		B		inner	BB005	JP7-ch1l/JP9-ch5l
23 24	20	S S		N N	inner inner	BB005	JP7-ch1h/JP9-ch5l
25 26 27	21		T T T		outer outer outer	BB005	JP3-ch2l/JP5-ch6l
28 29 30	22	N N N		S S S	outer outer outer	BB005	JP3-ch2h/JP5-ch6l
31 32 33	23		B B B		outer outer outer	BB005	JP8-ch3l/JP10-ch7l
34 35 36	24	S S S		N N N	outer outer outer	BB005	JP8-ch3h/JP10-ch7l

13 BBC-DSM tree

The basic idea is to compare the TAC and the ADC value individually for each PMT of the small tiles. There are four layer0 DSM boards for the small tiles with 8 ADC and the 8 corresponding TAC channels. Two cables connect each layer0 board to the layer1 DSM board (making the max of 8). The large tile have a separate DSM board in layer1. Large and Small tiles are combined in the Vertex DSM.

+ TAC and ADC are compared for each PMT separately. Only ADC values with a good TAC make it into the small-tile ADC sum. Only TACs with ADC values above threshold make it into the fastest TAC race.

. One DSM has 8 input channels, that means each DSM has all PMTs of two quadrants (Top(3) + North(5)) or (Bottom(3)+South(5)) The FPGA code is the same for the layer1 DSMs BB001/2/3/4.

+ We have separate bits for each of the 16 sub quadrants (East/West)*(Large/Small)*(Top/Bottom/North/South). The small tile quadrant hits go into the scaler, the large tile quadrants can be used in the topology (UPC) trigger.

. Large and small tiles are treated as separate detectors. There are only 16 bits available from BB101 and BB102 to the Vertex DSM, the possibilities to combine large and small tile information separately for East and West are limited. Specifically, we do not have:

  – a total ADC sum large(no TAC)+small(cleaned by TAC). There are only separate bits for large> threshold and small > threshold.
  – a combined Quadrant-hit-pattern separately for East and West which combines small and large tiles, i.e. T/B/S/N - (Large OR Small).

. Thansk to Jack, the (modified) DSM tree is cabled and documented.

14 Definitions

Quadrants: consist of either 3PMTs(Top/Bottom) and 5(North/South). The Tile/PMT numbers are swapped for East and West so that East:North=West:South and vice versa.

Small Tiles: #1-18; Large Tiles: #19-36
Inner/Outer Ring within the small or large annulus:
Small Tiles: Inner #1-6; Outer #7-18
Large Tiles: Inner #19-24; Outer #25-36

Clock Cycles: There are four DSM board clock ticks per RHIC cycle. First and Fourth are needed to latch in/out the data. The second and third are available for calculations. We can have one major serial operation per clock cycle on a channel. But one can have them in parallel on different channels. These major operations are for instance: Comparison to a threshold, Summing 8*8 bit numbers, (summing 16*8 bits has to be done in stages taking two cycles), Combining several and/or operations.

Override Cycles: To extend the limit of 2clock cycles we can use override cycles. Here we use two RHIC cycles to process input. The output of cycle 'n' corresponds to input from 'n-1'. This needs to be done in all DSMs in the affected layer of the tree (e.g BB001/2/3/4) to keep the output of that layer consistent. We have enough empty synchronization/delay cycles in the trees to implement this. An example for using overrides is the killer bit algorithm, where the condition if a hit in cycle 'n' may be good is calculated in cycle 'n-1'. Nevertheless, this is a more involved and messy feature.

Max TAC The TAC is the time difference of the leading edge of the signal to a common STOP signal. Therefore, the fastest(earliest signal) is the largest number =Max TAC. Each Tac value has a quality bit, i.e. the corresponding ADC value is above a certain thrshold. Only TACs with the quality bit set are compared.